Software Testing and Verification

Problem Set 5: Axiomatic Verification

1. Consider the assertion of *weak* correctness: {z<0} s {y=z+1}. Which of the following observations/facts would allow one to deduce that the assertion is FALSE and which would not? Consider the observations individually and briefly justify your answer for each.

    a. When the initial value of z is 3, the value of y is 4 when s terminates.
    b. When the initial value of z is -1, the value of y is 17 when s terminates.
    c. When the initial value of z is -3, the program does not terminate.
    d. When the initial value of z is -17, the value of y is less than the value of z when s terminates.
    e. When the initial value of z is -17, the value of y is less than the initial value of z when s terminates.
    f. The program s is: y := z+1
    g. The program s is: if z <= 0 then y := -3; z := -4 end_if
    h. The program s is: z := 5; y :=6; prod := z∗y

2. Consider the assertion: {x>y}

```
  temp := x
  x := y
  y := temp
  if temp>z then
    y := z
    z := temp
    if x>y then
      temp := x
      x := y
      y := temp
    end_if
  end_if
{x≤y≤z}
```

   Prove the above using appropriate RULES OF INFERENCE. Show all steps.

3. Prove the following assertion using the While-Loop Rule of Inference. Show all steps.

{N≥1}

```
Found := false
Index := N
while (Index>0 & (not Found)) do
  if Key=List[Index] then
    Found := true
  else
    Index := Index-1
  end_if_else
end_while
```

{(Found ∧ Key=List[Index]) V
(~Found ∧ ∀ 1≤ i ≤ N • Key ≠ List[i])}

4. Prove the following assertion using a suitable Rule of Inference for the Repeat_Until-Loop. Clearly state the Rule of Inference and show all steps. (Hint: Do NOT include P =>I as an antecedent in your rule.)

{N≥1 ∧ iorder} (where iorder = ∀ 1≤i<N • List[i]≥List[i+1])

```
First := 1
Last := N
Found := false
repeat
  Index := (First + Last) div 2
  if Key=List[Index] then
    Found := true
  else
    if Key<List[Index] then
      First := Index+1
    else
      Last := Index-1
    end-if-else
  end-if-else
until (Found or First>Last)
```

{(Found ∧ Key=List[Index]) V (~Found ∧ ∀ 1≤i≤N • key≠List[i])}

5. Consider the program below, where x and y are integer variables.

```
while (y<0) do
   y := y+x
   if (x≤0) then
      x := x+1
   end_if
end_while
```

a. For what initial values of x and y will the program terminate?

b. Can the Method of Well Founded Sets, as stated in class, be used to prove the program will terminate? If so, use it to do so. If not, suggest a generalization to the method that would allow its use in such cases.

6. Consider the following HYPOTHESIZED rules of inference for the "while" construct:

a.
$$\frac{P \Rightarrow (\sim b \wedge Q)}{\{P\}\ \text{while b do s}\ \{Q\}}\ ?$$

b.
$$\frac{\{P \wedge b\}\ s\ \{I\},\ \{I \wedge b\}\ s\ \{I\},\ (I \wedge \sim b) \Rightarrow Q}{\{P\}\ \text{while b do s}\ \{Q\}}\ ?$$

For each of the above, clearly indicate whether or not the rule is **valid.** If valid, provide an assertion of the form {P} while b do S {Q} for which it *could* be used. If not valid, prove this by providing a counterexample.